ONR Final Report
Development of Scenario Tutors in a Generalized Authoring Environment: Feasibility Study

19980126 100

Douglas M. Towne

Behavioral Technology Laboratories
University of Southern California

Sponsored by:
Office of Naval Research

Under ONR Contract No. N00014-97-1-0739

USC
UNIVERSITY OF SOUTHERN CALIFORNIA

Final Report
Contract N00014-97-1-0739

Development of Scenario Tutors in a Generalized Authoring Environment: Feasibility Study

Douglas M. Towne

Technical Report No. 119

Behavioral Technology Laboratories
University of Southern California
250 No. Harbor Drive, Suite 309
Redondo Beach, CA 90277

(310) 379-0844

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1998	FINAL TECHNICAL (5/15/97-12/30/97)

4. TITLE AND SUBTITLE
Development of Scenario Tutors in a Generalized Authoring Environment: Feasibility Study

5. FUNDING NUMBERS
N00014-97-1-0739

6. AUTHOR(S)
Douglas M. Towne

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Southern California
Behavioral Technology Laboratories
250 N. Harbor Dr., Suite 309
Redondo Beach, CA 90277

8. PERFORMING ORGANIZATION REPORT NUMBER
Technical Report No. 119

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release:
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

An analysis of a representative set of scenario domains indicates considerable congruence of the entities, goals, and functions involved, suggesting the feasibility of producing a generalized authoring environment capable of producing highly customized and domain-specific training systems. Such an environment would only be effective if a training system developer can easily assemble the domain-specific appearance and behavior of the entities in a particular application. A provisional design of a generalized authoring environment specifies the content and information required of the author and the built in instructional elements that operate upon the domain representation.

14. SUBJECT TERMS
Scenario tasks, Intelligent tutoring, Authoring environments, Training systems, Command and control, Simulation

15. NUMBER OF PAGES
38

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

ABSTRACT

An analysis of a representative set of scenario domains indicates considerable congruence of the entities, goals, and functions involved, suggesting the feasibility of producing a generalized authoring environment capable of producing highly customized and domain-specific training systems. Such an environment would only be effective if a training system developer can easily assemble the domain-specific appearance and behavior of the entities in a particular application. A provisional design of a generalized authoring environment specifies the content and information required of the author and the built in instructional elements that operate upon the domain representation.

ACKNOWLEDGEMENTS

The work described here was supported by the Office of Naval Research, under ONR contract No. N00014-97-1-0739, Susan Chipman Scientific Officer.

Don Thomas Sine served as consultant on the project.

A number of individuals were most helpful in providing information about existing systems and technology, including Milton Stretton and James McCarthy, Sonalysts, Inc.; Paul Radtke, NAWC-TSD; Sandra Marshall, SDSU; David Wilkins, University of Illinois; Menachem "Kemi" Jona, Institute for Learning Sciences; Wallace Wulfeck and Sandra Wetzel-Smith, NPRDC; Sharon Garcia, Armstrong Laboratory; Brian Minsk, Search Technology.

TABLE OF CONTENTS

List of Figures

List of Tables

Development of Scenario Tutors in a Generalized Authoring Environment: Feasibility Study

I. Background

A substantial number of tasks performed in military and civilian settings may be termed 'scenario' tasks. To achieve the broadest possible viewpoint we will include under that term any task that meets just two criteria:

1. The task is essentially one of making conscious decisions concerning the course of action to take at any time (dynamic decision making); and
2. The environment may change as a result of the actions taken by or on behalf of the decision makers.

Many scenario tasks happen to involve high risk of loss, either to human life or property, primarily owing to the responsive nature of the environment to the actions taken. One consequence of this risk is stress. In addition, many scenario tasks necessarily involve multiple decision makers and action takers. Example scenario tasks include these:

- conducting a fire fighting operation;
- directing air traffic at a civilian airport;
- commanding a military mission;
- providing emergency medical treatment;
- responding to a natural disaster;
- dealing with a terrorist operation.

In some of these cases the task is initiated by an unexpected event, thus the initial objective is to minimize loss of life or property. In other cases the task is conducted in an ongoing fashion to proactively achieve desired outcomes and to avoid undesirable outcomes. In some cases the decision maker is confronting natural forces that are difficult to predict; in other cases the opposing force is human. Typically, scenario tasks present a complex mix of forces to be overcome, options for planning and predicting outcomes, and uncertainties about the state of affairs and intentions and capabilities of others.

It so happens that most scenario tasks also involve the kinds of skills termed *high-performance skills* by Schneider (1995), viz., 1) they require more than 100 hours of training, 2) they result in easily identifiable differences among novice, intermediate and expert skill levels, and 3) they generally result in a high failure (washout) rate in progressing from novices to fully trained individuals.

Assessing individual or team performance in complex scenario environments is particularly difficult, owing to the fact that the outcome of the scenario usually cannot be judged as an absolute property. Thus, the fact that particular fire caused four fatalities and 20 million dollars in damage is insufficient to determine the proficiency with which the fire was fought. Likewise, the fact that a friendly aircraft was fired upon, or even struck, does not in the absolute tell us if this grievous outcome was a result of shortcomings in human performance, or in system design.

This issue is even more troublesome when evaluating the capability of a system— the hardware, the software, and the procedures — to carry out its intended task. It is extremely difficult to know in advance if a particular system design is susceptible to catastrophic performance failures under some conditions. Moreover, after-the-fact critiques of particular events often reveal profound differences in assessing system performance. Following catastrophic fires in Malibu, California, some homeowners, whose homes were knowingly left to burn to the ground, deemed the fire fighting operation a terrible failure—not a surprising assessment. The fire management officials countered by claiming that loss was minimized by moving the fire equipment and personnel to a position that allowed greater chance of stopping the progress of the conflagration. The only thing that is clear, therefore, is that performance can only be assessed in terms of the opportunities the situation affords for various acceptable outcomes, via expert decision making. Not surprisingly, it has been found (Kaempf, Wolf, Thordsen, & Klein, 1992) that expert tactical decision makers tend to seek a satisfactory course of action, not the optimal one.

Training Scenario Tasks

Training decision makers to perform scenario tasks is both critically important and difficult. Consequently, both military and civilian services have made huge investments developing training and certification systems. Some such training systems present all the components found in the field environment, as in a full-scale military exercise. In other cases the cost, time, and risk to deliver instruction is drastically reduced by simulating some aspects of the real world. Even in the latter case, there is typically heavy dependence upon highly skilled human instructors, and there is a very high cost associated with developing and delivering the instruction.

Clearly, resources that could improve training of decision makers will have great benefit. Possibly, the same or related tools could also contribute to an analysis of a current or proposed system design that would make explicit the possible outcomes under various conditions. These conditions could include an analysis of outcomes under different error conditions, as well as analysis of outcomes when the decision makers perform their jobs as expertly as possible.

Dependence Upon Human Expertise

Cost is just one limiting factor in relying upon human expertise to deliver instruction in this domain. Expert instructors simply may not be available in the numbers necessary to train novices, evaluate and remediate previously trained individuals, and to still perform their primary functions. Even when available, an expert may have considerable difficulty establishing training conditions that will yield productive learning experiences and evaluating ensuing performance. When there are multiple decision makers, multiple action takers, multiple and interacting objectives, and rapidly evolving situations, the task of monitoring and remediating performance may be overwhelming. This is particularly true because the instructor must consider both the process and the product of the learner's performance.

If a particular scenario task is short in duration and presents just a single objective, then one could effectively present a large number of such tasks to the learner and simply determine the outcome achieved on each trial. Since most scenario tasks are both lengthy and multi-faceted, instruction must inform the learner how particular decisions, made at particular times, affect the problem environment and the ultimate outcomes.

Automated Instruction of Scenario Tasks

As computer capabilities have grown so have the training functions that can be addressed by automated means. Unfortunately, while the cost of computer power is dropping dramatically, the costs to develop computer-based training or computer-augmented training are primarily labor-driven, and the complex nature of scenario tasks leads to exceedingly high development costs.

If it were not for the availability of extremely powerful programming languages and reusable high level display and analysis functions, these development costs would be far greater than they are. It is now timely to ask if there are not other, higher level, software resources that could be developed to harness development costs. More specifically, we ask if it is feasible to develop a general purpose authoring resource that might be used to produce highly customized training systems for a significant range of scenario task domains.

This report will address that question. Section II will explore the range of tasks that falls under the scenario classification; section III presents a survey of some scenario training systems and training aids for scenario tasks; section IV presents a generalized conception of scenario tasks; section V considers the training and authoring requirements imposed by scenario tasks; section VI presents a preliminary design of a generalized scenario trainer; and the report ends with conclusions in section VII.

II. Domain Specification

This section is concerned with establishing the range of tasks that fall under the scenario classification. It will outline the domain of scenario tasks without regard for the extent to which they might be amenable to training via a generalized system. While certainly not exhaustive, the resulting task list will include sufficient variation to assess the problem requirements.

Figure 1 provides a high level, non-exhaustive, itemization of some significant scenario tasks.

Military
    tactical command and control
        air mission or maneuver
        ground mission or maneuver
        surface sea mission or maneuver
        subsurface mission or maneuver
        damage control
    strategic
    logistics command and control
        battle management
        peacetime supply
Civilian
    air traffic control
    medical emergency treatment
    control of communicable diseases
    police operations
    911 dispatching
    natural disaster / rescue
    toxic disaster
Terrorism prevention / response
    hostage
    explosive
    biological/chemical/nuclear

Figure 1. A List of Representative Scenario Tasks.

In this listing terrorism is itemized separately, as it seems to transcend the military and civilian categories. Many possible task combinations are possible, such as earthquake plus fire, ground and air tactical mission, and police operation combined with a medical emergency.

From this list of representative scenario tasks, we proceed to a consideration of the general character of scenario tasks.

Character of Scenario Tasks

Scenario tasks fall in the center of a continuum which represents the degree of linkage between the decision process (as evidenced by actions) and the passage of time. As suggested in Figure 2, *sequential decision* processes are weakly linked to time — the decision process simply follows some ordered progression over time. In these types of tasks the time at which a decision is made does not materially affect the task environment, even if the decision is made under time stress. Examples of such tasks are depot-level fault diagnosis[1], stock portfolio management, and computer programming.

[1] In the depot the fault is not directly threatening systems or troops, thus task performance is not tightly linked to the environment.

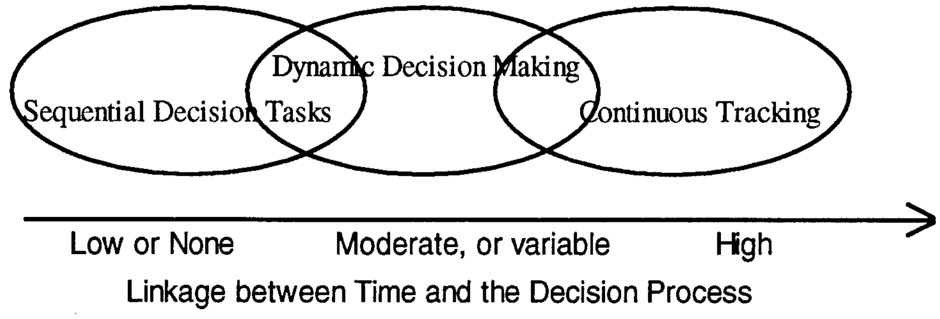


Figure 2. Types of Decision Tasks According to Linkage to Time.

At the other end of the continuum are continuous tracking tasks, which are certainly dynamic activities, but there may be few or no conscious and discrete decisions or actions. Furthermore, the action taken at any time is closely related to the time at which it is performed, and the condition of the system at any instant is directly linked to the time at which each action was performed, as well as the nature of the action.

Between these two extremes lies the entire landscape of dynamic decision making tasks, in which decisions and associated actions are relatively discrete, the decision at any instant may or may not be a function of the time at which it is made, and the system status may or may not respond directly to all actions taken. The scope of dynamic decision making tasks ranges from those in which the decision maker is primarily watching for the occurrence of some event or situation, called *attention* tasks, to 'speeded' decision making (Hunt, Joslyn, & Sandquist, 1996) in which the pressure to act quickly is a fundamental part of the task. Most dynamic decision making tasks offer a number of time windows within which the decision maker has relative freedom to consider alternatives, but the freedom may diminish with poor decisions early on. A rigorous treatment of this issue may be found in Rothrock (1995).

Of course there are no clear boundaries between these classifications. Operations such as space vehicle docking may involve early phases in which operational decisions are primarily sequential, intermediate phases in which decisions are only constrained by short-term time limits, and later phases in which the operator is engaged in a continuous tracking function in which the system status and the operator actions are tightly related.

A significant portion of the research on scenario tasks has been devoted to one domain, tactical decision making (TDM). Tannenbaum, Beard, and Salas (1992) offer the following attribute list to characterize TDM:

- rapidly evolving scenarios
- time compression
- threat
- adverse physical conditions
- auditory overload/interference
- high workload
- ambiguity
- command pressure

This characterization applies very well to the more general class of scenario tasks. The following table displays some judgments about the extent to which various scenario tasks involve these features:

Characteristic	tactical decision making	air traffic control	fire fighting	emergency medical care	disaster response	space mission control
rapidly evolving scenarios	+++	+++	+++	+++	+++	+++
time compression	+++	+++	+++	+++	+++	+++
threat (to a resource)	+++	+++	+++	+++	+++	+++
adverse physical conditions	varies	+	+++	++	+++	+
auditory overload/interference	+++	+++	++	+++	+	++
high workload	+++	+++	++	+++	+++	+++
ambiguity	+++	+	+++	+++	++	+
command pressure	+++	+++	++	++	++	++

+++ high involvement ++ medium involvement + low involvement

Table 1. Characteristics of Some Common Scenario Tasks.

While these assessments are entirely subjective, it does appear that all these tasks share these characteristics. Moreover, all of them are usually conducted as team enterprises, which partially accounts for the high workload, ambiguity, and auditory overload.

III. Survey of Some Scenario-based Systems

This section will explore some existing and emerging scenario-based systems for the purpose of identifying the instructional functionality, situational fidelity, authoring features, and the domain range of those systems. Many of the systems described here are not truly training systems. Some are simulators and some are simulators with decision support capabilities. Nevertheless, these systems function in the scenario task world, they were designed to aid training in some fashion, and their requirements for presenting scenarios are very relevant to this study.

Since an extensive review of the state-of-the-art in scenario-based training has recently been completed by Stetton and Lackie (1996), the first part of this section will summarize the findings of that study that are pertinent to this paper. The section will then present brief summaries of some systems and some pertinent ongoing research.

Stretton and Lackie Review

Stretton and Lackie's review gathered data on nearly 200 characteristics for each of 23 simulation-based and scenario-based training systems (Appendix A), all of which were ".. tools for training or for supporting training." The issues addressed by the review included authoring methods, training methods, simulation capabilities, data collection, and measures of performance.

A key finding was that the majority of the systems studied could ".. initiate, run, and control scenarios...", but that there were serious shortcomings in the authoring of scenarios and the training methods used, if any. Some of their specific findings are these (Stretton and Cannon-Bowers, 1996):

- Three of the 23 systems developed scenarios semi-automatically, but only for initial conditions.
- Two systems assisted in the development of scenario products in addition to the script.
- Many of the systems were of limited use in scripting a scenario.
- Human-computer interfaces for full scale training systems were very difficult to use for building a scenario.
- Six of 23 systems provided a means of assessing performance with system assistance.
- Only 2 systems provided feedback outside of replay that was accessible within a reasonable time.
- One of 23 systems formally linked training objectives to events in the scenario in software.

One of their explicit conclusions was that

> **"... extensive shortfalls were identified in the areas of training objective identification and traceability, scenario pre-training materials generation, and stage setting for scenario-based training tools for use by the CSTT (combat system training team) to create and manage training on the ship."**

The study also looked at important issues beyond the authoring, simulation, and training capabilities of the systems. Among these were the following:

- the host platform and operating system employed
- connectivity with other systems and data bases
- adaptability of the training to available manpower and system resources

The review showed that there was not a system available that provided adequate scenario authoring, performance evaluation, and training functionality, to meet the needs of combat systems team training and anti-air warfare team training. The Training Management Module (TMM) (Stretton and Johnston, 1996) has been designed to meet those needs, particularly those related to pre-training materials generation. The design of TMM also represents an excellent baseline with which to analyze the requirements of a generalized authoring and training system.

Existing Training Systems

We turn now to a view of several existing training systems, for the purpose of identifying the objectives and processes of these systems and the challenges they would present to a generalized authoring system.

GT-AAWC. The Georgia Tech AAWC Simulation Suite (GT-AAWC) (Kirlik, Fisk, & Walker, 1996) was developed at the Human Attention and Performance Laboratory of the Department of Psychology, Georgia Institute of Technology. It was designed to support experimentation and evaluation of performance in tactical decision making, specifically the AAWC (Anti-Air Warfare Coordinator) position in CIC. The software suite includes tools for 1) authoring the air and sea entities and events of a CIC scenario, 2) running a real-time, event-driven of the Aegis CIC environment, 3) part-task training of the operation of the simulation platform, and 4) analysis of individual performance.

The simulation component is termed AAWC Simulation Platform (GT-ASP), which is very similar to the DEFTT system but it addresses individual performance rather than team performance. Of particular interest are tools for assessing task performance. These measure such individual variables as the time to identify a track and time to issue a warning, and they sense violations of rules of engagement, nine of which were formalized. Two example rules of engagement are as follows:

1. If a hostile aircraft moves to within 50 miles of one's own ship then a particular warning should be issued to that aircraft.
2. If a hostile aircraft moves to within 30 miles then the fire control illuminator should be turned on.

To be correct, an action must be both timely and accurate. Timeliness of an action is determined by computing a *window of opportunity* (Rothrock, 1966) for it, i.e., the earliest and latest time at which it could be correctly performed, and determining if the action was performed in that interval. Combining the three timeliness categories of an action (early, on time, and late) with its two correctness categories (correct, incorrect) yields six possible outcomes. In addition there are three other categories for commission, omission, and correct no action, for a total of nine possible evaluations of an action.

The part task training capability of this system has been experimentally applied to two components of the AAWC task: 1) manipulation of console controls, and 2) interpretation of displayed symbology. The rationale for selecting these two aspects of the task is that they compete for the operator's attention and consume valuable time if they are not performed automatically and simultaneously with higher level decision processes. The presentation of part-task training is done via a series of discrete exercises, rather than within the context of the full scenario simulation.

The system has been used to experiment with various feedback schemes. In one, the operator is provided feedback, in the form of a prioritized list of objects in the scenario that deserve action. The list identifies the object (track number, bearing, and range) along with the cue that identifies the rule that applies, .e.g., "identify". In essence, this list presents an expert's determination of what should be done next.

DDT/LC2. Brecke and Garcia (1995) developed a training methodology for the quintessential logistic decision making task, the allocation of resources to operational units during battle conditions. This methodology is delivered in the Desktop Decision Trainer for Logistics Command and Control (DDT/LC2).

In this methodology, an exercise consists of three phases:

1. orientation, in which the novice battle manager receives a "shift briefing" which provides the latest information about supplies, transport systems, and battle conditions. The information resources are briefings, orders, status boards, and message logs.

2. operations, in which the learning logistician receives messages, some of which are demands for resources from the field, some are replies to messages sent out previously, and others convey changes in the status of supplies, plans, transportation, and so on. The demand messages present the need to respond in some manner (refuse the demand, meet the demand, delay response, partially meet the demand, etc.). Several types of instructional feedback or assistance are possible during this phase, including a) immediate evaluation and correction of decisions, b) 'natural' feedback which is the change in the simulated world, partially caused by the decision of the learner, c) suggested solutions, d) prompts to direct the learner's attention to important issues, e) and 'exploratory optimization', which allows the learner to offer a decision for analysis by the system, which could then be executed or withdrawn.

3. debriefing, in which the instructional system evaluates the learner's performance on a just completed exercise.

Courses are organized into levels, the first of which presents the simplest possible problem (the epitome level), and subsequent levels increase the complexity, the uncertainty, and the time constraints of the decision problem. At each level, the learner progresses from novice to expert.

<u>DEFTT</u> – Decision Making Evaluation Facility for Tactical Teams. DEFTT is a testbed system for conducting tactical decision-making research. It is currently used to conduct experiments for the Tactical Decision-Making Under Stress (TADMIS) program. The simulation component, the Tactical Advanced Simulated Warfare Integrated Trainer (TASWIT), simulates an Aegis cruiser Combat Information Center (CIC). The research configuration is comprised of six consoles and an Experiment Control Station (ECS), all electronically linked. In addition there are voice communications presented on the four audio channels, including voiced messages from human role players.

The experiment participants, the Commanding Officer and the Tactical Action Officer (TAO), view and operate a Large Screen Display. The other consoles are manned by skilled operators who perform their positions as they normally would, including responding to orders issued by the CO and TAO. Performance data is recorded on a multichannel recorder, along with voiced messages, and optionally video.

The displays, consoles, communications, and task environment are extremely realistic, with the one exception that the simulated aircraft and ships do not respond to orders issued by the participants. A relatively small number of highly complex scenarios are employed, representing peace keeping missions (i.e., non-engagement situations) in the Persian Gulf.

TADMUS Decision Support System (DSS). In addition to the Aegis CIC consoles, the DEFTT facility supports the experimental Decision Support System, developed to enhance decision making performance. The DSS presents information about 1) a single track (Track Summary, Track Profile, and Response Manger), 2) historical track information in terms of threat assessment (Basis for Assessment, Comparison to Norms), and 3) a summary of information about all targets (Track Priority List, Alerts List).

Radar System Controller Intelligent Training Aid (RSC ITA). The RSC ITA (McCarthy, Pacheco, Banta, Wayne, and Coleman, 1994), also now known as RITA, is a simulation-based training aid that supports individuals learning to become skilled radar system controllers (RSCs), in a master/apprentice environment. Superficially, the task is one of monitoring a PPI and other displays and operating a console to control a phased array radar system. The decisions about what to monitor and how to employ the search radar are ones which require considerable training, as they involve tradeoffs between time, certainty, allocation of radar resources, and the nature of the tactical situation in progress.

The student model is based upon a hierarchy of learning objectives, high levels of which correspond to the more difficult decision making functions of an RSC, lower levels corresponding to the more mechanical operation of the equipment. The sources of evidence for and against mastery were then specified, in terms that could be applied during the exercise, and an instructional expert was developed to intervene when the student model reflected a sufficiently critical need. Interventions are considered by the expert following certain discrete and identifiable learner actions. When presented, instructional interventions employ a *fading* technique in which the amount of remedial information is related to the extent of proficiency deficit exhibited by the learner. The system does not incorporate a scenario authoring capability, but the embedded domain expert has the capacity to re-present portions of existing scenarios to meet ad hoc needs of the learner, and, according to its developers, has the potential to modify aspects of those parts being presented in a remedial phase.

Interactive Multisensor Analysis Training (IMAT). IMAT is designed to promote, through the generation of enhanced visualizations, understanding of the complex interrelationships among oceanographic phenomena, shipboard sensors, processors, and displays. Thus it represents a resource that could be employed within a training system. IMAT employs high-end computer graphic technology to present representations that are linked to detailed simulations of sonar consoles and displays, environmental attributes, oceanographic data, and physical phenomena. The intent is to enable SONAR operators to better understand the processes that underlie the propagation of signals under water and to better interpret the displays that represent those signals.

PC-IMAT. An adaptive tutoring system using IMAT technology, PC-IMAT, is under development. The focus of the work is on identifying critical indicators of conceptual deficiencies in understanding the effects of oceanographic and environmental factors on acoustic propagation. The indicators will be identified by comparing performance of experienced sonar technicians to that of novice technicians, by interviewing instructors, and by analyzing performance data from IMAT classrooms. Two basic instructional strategies are being explored: 1) presentation of exercises based upon a student model (SMART), and 2) complete learner control.

TRIDENT Command and Control Team Trainer (CCTT). The CCTT replicates the 8L(T) periscope platform, the ship control station, the AN/WLR-8(V)5, and several plotter tables in the TRIDENT control room, and it simulates the MK48 Torpedo, MK57 MOSS, functions of the M96, and associated communications systems. Ocean environmental effects, targets, passive and active sonars, and ESM emissions are simulated. The trainer can operate in a stand alone mode or, when coupled to the Sonar Team Training Laboratory (STTL), in a team training mode. The trainer operates in run, freeze, restart/playback, and secure modes.

Submarine Combat Systems Team Trainer Device 21A43. This system is used to train fire control equipment operation, submarine approach, attack and surveillance, and combat system team coordination. The system incorporates components of the MK-1 Fire Control System and simulations of other tactical equipment.

TRIDENT MK-118 Defensive Weapon Subsystem Operator Trainer (DWSOT). The DWSOT allows instructors to generate and initiate scenarios of simulated missions involving the DWS Fire Control System, to change scenario parameters, and to monitor learner performance during exercises. The simulation includes data for ownship, targets, sensors, weapons, and the ocean environment.

Run. Run (Jona & Kass, 1997; Shank & Korcuska, 1996) is one of five[2] 'teaching architectures' (Shank, 1994) developed to provide an environment in which a learner can become engaged in a simulated problem situation, primarily via video segments. The student's task is to manage a complex, dynamic system and to react to unexpected events that threaten the stability of the system. Two applications developed within this context are: 1) Fire Commander, in which the student is responsible for directing the fighting of a house fire involving people at risk, and 2) Zookeeper, in which the student deals with everyday zoo management issues as well as confronting unexpected problems (e.g., fighting gorillas). In both cases the student has access to domain-specific information and can explore the interrelationships that exist among the system variables. The environment offers defined decision points, at which the learner can access information, obtain advice, and enact a decision.

IV. A Generalized Conception of Scenario Tasks

Using the foregoing systems and their task environments as a basis, we now offer a generalized conception of scenario tasks. The objective is to derive a statement of a scenario task that embraces these systems and identifies the entities that would comprise a generalized training system.

[2] Eight architectures have been designed, five implemented.

Scenario Entities

A general scenario task is one which involves one or more decision makers and some subset of the following entities:

- some *asset* the decision maker is attempting to defend or protect;
- *information sources* who/which provide information of varying accuracy, precision, and immediacy;
- *resources* which the decision maker can bring to bear to accomplish the scenario objectives;
- *ordained* events, which are destined to occur during the situation;
- *cooperative* agents that work with, or respond to, the decision maker or team;
- *indifferent* agents that respond only to the physical environment;
- *hostile* agents that intelligently threaten the assets being protected or defended;
- terrain or spatial boundaries which may constrain movement of the decision maker and the agents;
- atmospheric conditions which may affect visibility, temperature, communications, etc.
- a *history* that expresses prior events and experiences that may affect expectations of the future.

<u>Decision Maker</u>. The decision maker or team performs actions during the scenario, and some of the agents in the scenario may react to those actions. Many actions are commands to cooperative forces, either to provide information about the situation or to exert physical measures on the hostile or indifferent agents. Other actions may involve communicating with friendly, indifferent, or hostile agents.

<u>Assets</u>. Assets are some valued element the decision maker or team is attempting to defend or protect. Most, if not all, scenario tasks involve some asset. The asset may be a set of individual entities of differing value.

<u>Information Sources</u>. These provide information about the situation to the decision maker. Some information is provided as it becomes available, other is provided upon request by the decision maker. Information can be of varying accuracy, precision, and immediacy.

<u>Resources</u>. Weapons, remedies, or other physical processes directed against hostile or indifferent agents to counteract or resolve them.

<u>Ordained Events</u>. These are unavoidable events or changes in conditions that will occur at some predetermined time into the scenario. Usually the occurrence of these events are not known to the decision maker at the initiation of the situation. Examples include wind shifts during fires (not predictable), failure of a communication system at some time after the scenario begins (not predictable), and loss of telemetry signals when a space vehicle passes behind the moon (predictable).

Agents. Agents are entities representing either forces to overcome, or resources to employ. Usually agents are human or machine. Agents fall into these three classes:

- *Cooperative agents*, who respond to commands if they receive them and they can respond. They may either be under the direct command of the decision maker, or they may obliged in a less formal way to respond to the orders or requests of the decision maker. Most cooperative agents are human beings or human-controlled entities, although an automated information system could be included in this class. In team tasks some agents may perform actions autonomously.

- *Indifferent agents* who/which only respond to physical action taken upon them. These are usually physical forces such as fires or viruses. Some indifferent agents, such as tornadoes, do not respond to any actions.

- *Hostile agents* usually do not respond to commands, and they will intelligently threaten the assets if possible. Hostile agents are human-driven or intelligent automated systems that can plan and execute attacks on the assets.

Atmospheric Conditions. These may affect or constrain the actions that can be taken, the information received, etc.

Example Task Descriptions

The scenario task specification outlined above is now briefly applied to the training resources surveyed previously.

Tactical decision making (GT-AAW, DEFTT/TADMUS, TRIDENT, etc.). The decision maker attempts to protect own forces and other friendly forces from possibly hostile agents by issuing commands and requests to cooperative agents. The commander may issue orders to the crew or friendly forces to provide additional information or to take physical action against hostile agents by employing weapon resources. The goals are to avoid attack, to avoid inappropriate action taken against others, and to maintain a safe condition around one's own forces.

Logistics decision making (DDT/LC2). The decision maker attempts to deliver resources to own forces with the goal of maximizing the effectiveness and safety of the forces. The decision maker must consider past and possible future actions of hostile agents to prevent delivery as well as the conditions created by various decision alternatives.

Radar employment (RCS ITA). The decision maker employs an information source to detect and identify hostile agents. The employment of the search radar must be done in a manner that is appropriate considering the tactical situation at hand. This includes trading off probability of detecting hostile agents with that of endangering own forces. Ocean floor terrain and other oceanographic factors may further complicate the task.

Fire fighting (Run-Fire Commander). The decision maker attempts to protect the assets of life and property being threatened by a fire, an indifferent agent. The decision maker's actions are commands to the units to supply information, to take physical action against the fire itself, using the resources of water or chemical weapons, or to perform other operations, such as removal of fuel or movement of personnel. The fire only responds to weapons employed against it, to atmospheric conditions, and to availability of fuel. The goals are to minimize loss of life and property at the conclusions of the situation, when the fire is extinguished.

Zoo management (Run-ZooKeeper). The decision maker monitors conditions of the zoo via various information sources, plans for ongoing operations, and responds to special problems. If problems arise, the decision maker must protect the safety of the visiting public as well as that of the animals.

Scenario Goals and Functions

Extensive cognitive analysis of the tactical decision making task (Klein, 1992; Miller, Wolf, Thordsen, & Klein, 1992) has produced a list of primary goal states of some critical CIC incidents, and the researchers note that these goal states may be relevant to other U.S. Navy missions beyond anti-air warfare. Table 2 lists these goals and functions.

No.	Goal	Function/Activity
1	Determine intent	CIC crew attempts to determine the intentions of a track, such as whether or not the track is hostile.
2	Recognition of a problem	Crew tries to determine if they are faced with a potentially threatening situation.
3	Take actions to avoid escalation	Crew takes deliberate steps to avoid the escalation of an incident into an engagement.
4	Take actions toward engaging	Crew takes preparatory steps needed to engage a track.
5	Monitor on-going situation	The CIC crew monitors a situation to detect any changes in the situation.
6	Identify track	Crew attempts to determine the identity (e.g., country of origin) of a track.
7	Allocate resources	The CIC crew attempts to allocate limited resources to deal with the current situation.
8	Prepare self-defense	Crew takes steps toward self-defense, such as bringing up the CIWS.
9	Conduct all-out engagement	Crew actively engages a track with a weapon system.
10	Monitor tracks of interest	Crew monitors a track which has some significance to the current situation.
11	Reset resources	The crew returns ship resources to pre-incident status.
12	Collect intelligence	CIC crew actively tries to collect information on a track.
13	Trouble-shoot	Crew tries to trouble-shoot a system.
14	Determine location of a reported track	CIC crew attempts to determine the location of a reported track.

Table 2. Goals, Functions, and Activities of a CIC crew.(Miller, et al., 1992)

It is not difficult to extend these elements beyond CIC or even beyond military domains. By substituting generalized scenario statement terms for the CIC-specific terms, we derive Table 3, which might apply to a wide range of scenario tasks (italics indicate substitutions).

No.	Goal	Function/Activity
1	Determine intent of agent	*Team* attempts to determine the intentions of an *agent*, such as whether or not the *agent* is hostile.
2	Recognition of a problem	*Team* tries to determine if they are faced with a potentially threatening situation.
3	Take actions to avoid escalation	*Team* takes deliberate actions to avoid the escalation of an incident into an engagement.
4	Take actions toward engaging agent(s)	*Team* takes preparatory steps needed to engage an *agent.*
5	Monitor on-going situation	The *team* monitors a situation to detect any changes in the situation.
6	Identify agent	*Team* attempts to determine the identity of an agent.
7	Allocate resources	The *team* attempts to allocate limited resources to deal with the current situation.
8	Prepare self-defense	*Team* takes steps toward defense of own forces.
9	Conduct all-out engagement	*Team* actively engages an agent with a counteracting *resource.*
10	Monitor agents of interest	*Team* monitors an agent which has some significance to the current situation.
11	Reset resources	The *team* returns system resources to pre-incident status.
12	Collect intelligence	*Team* actively tries to collect information on an *agent.*
13	Trouble-shoot	*Team* tries to trouble-shoot a system.
14	Determine location of reported agent	*Team* attempts to determine the location of a reported agent.

Table 3. Generalized Goals, Functions, and Activities of a Scenario Task.

The table suggests that there may be considerable congruence among widely different scenario tasks, in terms of the goals, functions, and entities involved. To further analyze this, Table 4 presents the manner in which these goals and functions might apply to three non-tactical scenario tasks.

No.	Goal	Damage control	Medical emergency	Air traffic control
1	Determine intent of agent	Not applicable	Not applicable	Determine intent of aircraft
2	Recognition of a problem	Call for initial damage reports	Assess patient's vital signs	Scan PPI for critical problems
3	Take actions to avoid escalation	Act against most threatening conditions	Attempt to stabilize patient	Issue directives to increase separation
4	Take actions toward engaging agent(s)	Alert personnel to prepare crew and equipment	Prepare emergency facilities and personnel.	Not applicable
5	Monitor on-going situation	Monitor status of damaged system	Monitor patient vital signs and other test results	Monitor PPI for current status of aircraft
6	Identify agent	Determine source of system problems	Determine cause of abnormal symptoms	Identify displayed aircraft (destination, airline, type)
7	Allocate resources	Direct personnel and equipment as needed	Employ appropriate personnel and equipment	Manage airspace, landing areas, radio frequencies
8	Prepare self-defense (or asset defense)	Prepare for impending effects of damage	Anticipate impending patient reactions & needs	Project future airspace configuration and needs
9	Conduct all-out engagement	Assign all necessary resources to damage control	Treat critical patient with all available resources.	Utilize all necessary resources to avoid collision or land aircraft
10	Monitor agents of interest	Focus on most threatening damage areas	Focus on most serious symptoms	Focus on most critical aircraft and airspace sectors
11	Reset resources	Withdraw personnel and equipment	Finish use of emergency facilities	Release use of emergency resources
12	Collect intelligence	Same as #5	Same as #5 and #6	Same as #6
13	Trouble-shoot	Deal with special problems	Diagnose puzzling test results	Respond to emergency call
14	Determine location of reported agent	Determine extent of fire, flooding, ...	Conduct tests to locate injury, bleeding site, etc.	Identify location or altitude of aircraft

Table 4. Goals, Functions, and Activities of Three Scenario Tasks.

As expected, the goals and functions applicable to tactical decision making appear to constitute a superset of those that apply to most other scenario tasks, since the tactical environment may involve a hostile agent.

While the degree of congruence among these few scenario tasks is comforting, a training system for any one of the tasks must also incorporate very specific and accurate domain-specific content and processes. An authoring system must therefore facilitate 1) authoring of domain-specific entities, content, and processes, and 2) mapping domain-specific matter into the generic classes, for processing by built in instructional functions. The prospects for doing this in a generalized environment will be considered next.

V. Analysis of Training and Authoring Requirements

Prospects for Scenario Generation

Several research programs are considering the process by which scenarios might be produced, either manually or automatically, to meet particular training requirements. While the majority of this work is being conducted in the tactical decision making arena, it is not particularly difficult to imagine the application of these methods to other domains.

Managing Stress Inherent in Scenarios

Hall, Dwyer, Cannon-Bowers, Salas, and Volpe (1993) have focussing on the problem of producing scenarios to train teams working under specified levels of stress. They employ various evaluation methods to assess the stress inherent in a given scenario and the level of performance of individual members of the team. The requirement is to identify key decision making tasks, and to determine the associated knowledge, skills, and abilities required to perform them. The identification of key tasks is done via interviews of expert CIC personnel. The Cognitive Network of Tasks (COGNET) (Zachary, Zaklad, Hicinbothom, Ryder, Purcell, & Wherry, 1991) methodology is then employed to determine the knowledge requirements of the AAW Coordinator. Next, SMEs assemble scenarios that would require action by all the team members, and would impose either moderate or high levels of stress. The scenarios are then analyzed in terms of the inherent workload and levels of ambiguity.

Workload is assessed by evaluating the key elements in a tactical scenario, the tracks. The process involves categorizing each track in a scenario into one of four categories: 1) unknown, 2) interest, 3) action, and 4) engageable, per detailed descriptions of each category. While the relative workload associated with each category is not known (and may well vary with other variables), the research of Zachery, et al. (1991) indicates that workload is least in tending to a category one track, and increases to level four.

Ambiguity is assessed by evaluating the events in a tactical scenario that involve vague, conflicting, and missing information. An event, such as loss of electronic emissions or receipt of a new communication, may impose additional workload to resolve ambiguity the event causes. The Ambiguity Assessment Matrix is a form with which these ambiguity sources are documented in textual form. While the research reports do not mention it, we might assume that events that reduce ambiguity could also be documented in this manner.

The final step in the scenario development and rating process is to have Navy SMEs specify the actions that each team member (CIC position) should take at each event in the scenario. This forms the basis upon which actual performance is measured. During or after a scenario exercise, trained observers rate the performance of individuals against the specified correct actions and action sequences.

Damage Control

David Wilkins and his group are studying the generation of damage control scenarios, and they have collected some 50 person-weeks' of preliminary data to support further work. For the initial studies a scenario is specified in terms of a primary damage event. A central goal of this research is to generate a scenario to meet a given objective. A key difficulty identified by the early work is that the actions of the decision maker may so alter the scenario that the objective may not be reached.

Shipboard Instructor Training and Support (SITS) Scenario Development

Stretton and Lackie (1996) have documented a functional architecture for development of tactical scenarios within the SITS program. The design of this specific scenario task trainer stands as an excellent baseline against which to consider a generalized authoring system, since 1) the domain, tactical decision making, involves nearly all the factors found in other important domains, 2) the architecture was produced in light of an extensive state-of-the-art review, as outlined previously, and 3) the objectives are ambitious.

The six key goals established for the SITS scenario development system are as follows:

1. Create realistic scenarios for shipboard training based on team and individual historical performance.
2. Clearly represent the linkage between missions, METLs (Mission Essential Task List), shipboard training objectives, performance measures, and scenario events.
3. Communicate the linkage to data collection agents, expert models, and CSTT (Combat System Training Team) cueing.
4. Create a scenario and supporting materials meeting the above goals in less than 4 hours.
5. Provide a scenario script file compatible with candidate scenario implementation systems.
6. Provide a logical and usable display and control architecture that meets the above goals and user requirements.

The preliminary design also calls for 1) the capability to control levels of difficulty and stress placed upon the individual operator, 2) tracking of both individual and team performance, 3) provision of on-line help, and 4) capability to accept and incorporate inputs that establish training priorities.

Instructional Challenges and Opportunities

The nature of scenario tasks greatly influences the opportunities and problems associated with their instruction. This section briefly outlines some of the issues that influence the design of an instructional system for scenario tasks.

Effects of errors. scenario tasks are often closely related to safety or human survivability, thus attaining very high proficiency is critical. This places high demand upon an instructional system, as it must be capable of presenting a wide range of situations, and it must deal intelligently with individuals who may already be considered highly skilled. The system must permit the learner to commit an error and see the results of the action, and the learner must be able to observe expert performance on the same problem.

Options for Interacting with the learner. Because dynamic decisions are relatively discrete and explicit, an instructional system can refer to the learner's actions or recommended actions verbally or graphically, and it can demonstrate recommended actions verbally or graphically. Furthermore, an instructional system can project the likely outcomes of alternative decisions and relate these outcomes to the learner, either during an exercise or following it.

Minimizing effects of instructional intrusions. Unlike continuous tracking tasks, scenario tasks typically permit a relatively wide range of instructional interactions to be presented within the context of the task. In some cases warnings or recommendations may be presented with minimal impact upon the instructional environment. In other cases, time may be artificially slowed or stopped, to permit presentation of instructional material before continuing. Similarly, time may be accelerated when desired, allowing a scenario to evolve rapidly past uninstructive phases of a situation. To minimize intrusion, a training system should be capable of discussing errors at the conclusion of an exercise, then presenting the scenario from the point at which a critical error was committed, allowing the learner to attempt an improved completion.

Repeatability of learner performance. Because decisions in scenario tasks usually result in discrete actions rather than continuous tracking motions, individual exercise performances can be saved to file readily and replayed, allowing the learner to review his or her decisions and the outcomes that resulted. This same replay capability also permits experts to perform the task and then to provide those performances as demonstrations of recommended procedures. In both cases, automatically generated critiques of performance can accompany replayed exercise performances.

Requirements for real time situation assessment. scenario tasks and continuous tracking tasks share the characteristic that any particular exercise is identical for different learners only at the very beginning of the exercise. Following the first or second decision, scenarios can evolve in a manner that is highly individualized. Sequential decision processes, such as diagnostic exercises, evolve in highly individualized ways too, but the character of any particular exercise, the fault, is the same for all learners throughout the exercise. In scenario tasks, a few poor decisions, including the decision to do nothing, can transform a trivially simple scenario into an entirely different problem, one that may involve high time stress and critical exposures to errors. Because of this, an intelligent instructional system for scenario tasks must be extremely powerful in:

1) assessing the status of the task environment at any instant,
2) sensing the causes of critical conditions,
3) generating recommended actions relative to the current situation, and
4) assessing the learner's proficiency to generate appropriate scenarios at each stage of instruction.

Team tasks. Most scenario tasks involve multiple performers. In some cases the learner's role in the team is the only one that makes operational decisions, but very often others do so as well. In special cases, such as the CIC simulation system (Towne, 1995), the performance of all other roles is done correctly and within expected time requirements. In the real world the performance of other team members can be incorrect or inaccurate, and it can be delayed to the point of affecting the operation.

To gain maximum benefit from an intelligent training system, it should be possible to simulate the performance of other team members at desired levels of skill and speed, permitting the learner to become proficient in monitoring and confirming the performance of others and in dealing with problems resulting from their performance.

VI. Preliminary Conception of a Generalized Scenario Tutor

This section will present a provisional conception of a generalized authoring and instructional delivery system for scenario training.

Components and Functions of a Generalized Trainer

The major components of the system (Figure 3) are 1) a scenario generation resource, 2) a simulation capability, and 3) an intelligent tutoring process. In the figure, elements that are authored for a particular domain are represented in dotted outline, while the built-in domain-general resources are shown in solid outline.

The Scenario Generator operates upon domain-specific scenario prototypes to produce a customized variant of the prototype that involves a desired level of difficulty, depending upon the current proficiency of the decision maker or team. The methodology for doing this is outlined below.

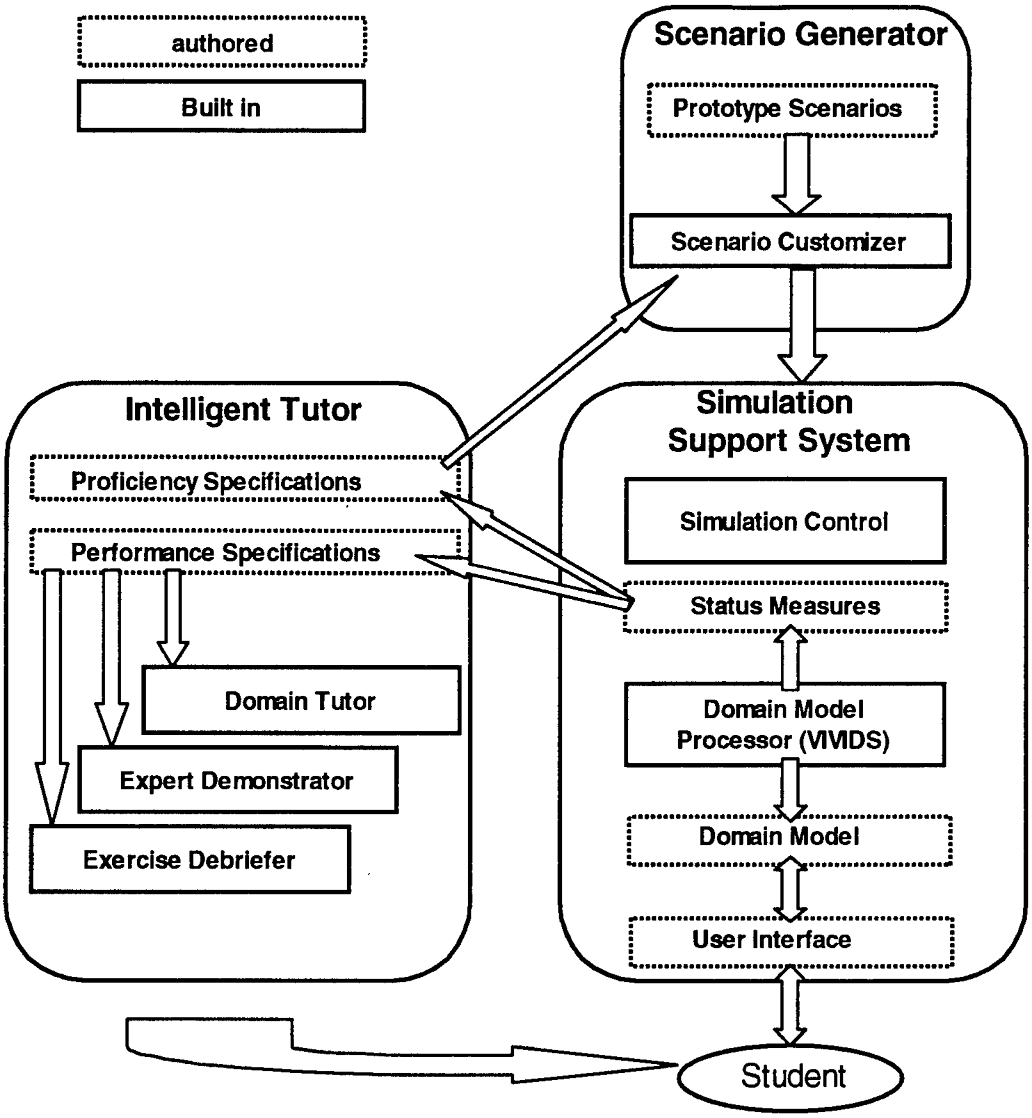


Figure 3. Top Level Design of a Generalized Scenario Tutoring System.

The Intelligent Tutor component includes domain-general processes for evaluating the learner's performance, for generating remedial instruction, for debriefing the learner after an exercise is completed, and for demonstrating expert performance. All of these processes operate upon authored domain-specific proficiency and performance specifications, the syntax of which is detailed in Appendix D. Performance specifications are declarations about how the task, or the scenario in the task, should be performed, i.e., what decisions should be made under specified conditions. The proficiency specifications yield positive or negative numbers as indications of the proficiency of the decision maker when a particular condition exists in an exercise.

The Simulation Support System consists of five elements:

1. domain-general simulation control functions for maintaining simulation time, freezing time, accelerating time, backing up a scenario to a previous time, and replaying completed exercises;

2. a domain-general simulation engine that maintains the attributes of the domain model (VIVIDS). This updates the values of expressions and executes processes in the domain model as time passes and as the student executes actions, and it updates those graphical elements of the user interface that reflect the status of the domain model.

3. domain model – an authored set of rules and processes that specifies how the domain changes over time, and how it responds to actions carried out by, or on behalf of, the decision maker. The model of a scenario task would be composed of objects representing agents, team members, resources, and assets. Rules and procedures would be authored to specify how each of these elements behave, in terms of such properties as intention, capabilities, limitations, and strategy.

4. status measures – expressions that reflect significant conditions of the domain model. These measures are referred to by the proficiency specifications embedded in the intelligent tutor. An example status measure might be the time since the decision maker last performed a certain kind of action, or the number of requests awaiting response.

5. user interface – a graphical and responsive representation of the domain which the student manipulates and observes. The user interface displays some or all of the agents, resources, and assets of the domain model. The user interface would include display views of the world available to the decision maker, controls for managing or changing the display, and controls for issuing commands or obtaining information.

Authoring Requirements

Each of the authored elements shown in Figure 3 would require a special-purpose authoring system, as detailed in Table 5.

Authored Element	Authoring Resources Required
User interface	Graphics editor
Status Measures	Rule editor
Domain Model	Rule editor, Event editor, Agent Editor, Team Editor, Resource Editor, Asset Editor, Event Editor
Prototype Scenarios	Scenario editor
Proficiency and Performance Specifications	Rule editor

Table 5. Authoring Resources Required.

Two of these editors, the Rule editor and Graphics editor, exist, in the VIVIDS authoring system. The Agent Editor would capture such properties as the intentions of the agent (hostile, friendly, indifferent), the agent's capabilities to assist or hamper the operation, and so on. The Team Editor would allow the domain author to specify the speed, accuracy, and error rates of various team members. The Resource Editor would allow specification of the weapons or tools available to the decision maker to combat hostile or indifferent agents. The Asset Editor would capture such properties as relative value of different resources to be defended or protected, in case tradeoffs must be made. The Event Editor would facilitate the scripting of within-scenario events in terms of event codes (e.g., windShift, airEmergency, or courseChange), time of the event, duration of the event, and values resulting at the end of the event (e.g., the final course or final amount of fuel onboard).

The Scenario Editor would permit the author to assign difficulty ratings to two existing scenario instances in order to automatically generate instances of intermediate difficulty, as outlined next.

Scenario Generation

While not infeasible, the approach of automatically generating domain-specific scenarios wholly from detailed specifications of the domain appears to be extremely difficult and costly in terms of authoring effort required. Furthermore, such an approach would be exceedingly difficult to generalize to any useful extent. The problem is that coherent and interesting scenarios could only be generated automatically via application of very extensive world knowledge about the domain.

A more workable approach is to develop authoring resources that permit rapid construction of scenarios representing just a few levels of difficulty, and to develop special routines that can customize those prototypes to produce an instance offering a desired level of difficulty. The following will outline the general approach that could be taken to construct and customize prototype scenarios. In this context, a scenario is some situation, such as "an airliner is low on fuel and requires assistance in landing", or "a fire has broken out in a chemical processing plant." An instance of such situations specifies all the speeds, positions, and other characteristics that make the situation fully defined.

Scenario Construction

The initial construction of a scenario is facilitated by permitting the author to position the objects of the scenario directly on the screen with a mouse and to key in pertinent variables that define the objects. Maneuvers or actions of the objects can be further specified by composing a script of actions, which is simply a formatted list of actions taken by all objects during the scenario[3]. Likewise, other changes in the domain, such as changes in the weather, can be included in the script.

---

[3] Other rules and specifications dictate the object's conditional responses under various situations. The events discussed here are those that will take place regardless of actions by the decision maker.

The domain expert will compose *two* instances of each scenario to be presented, one of which is the easiest version of that situation which will be presented, and one is the most difficult.[4] These two extreme cases, or prototypes, can differ from one another in terms of any of the task variables, such as different initial positions of the various agents, different characteristics of the agents or the personnel in the team, or the magnitude of any ordained events. In addition, the author will assign a *level of difficulty* to the two instances, say a number between 1 and 10.

This difficulty assessment may be made using any means, including subjective assessment, use of available student data, or performance scores achieved by an expert working the scenarios.

Thus, for an inherently difficult situation, the level of difficulty of its easiest instance might be 5 and the most difficult instance 9 or 10. Before proceeding, it is important to note that scenarios typically involve many factors that are unknown to the decision maker, and only unfold as the scenario progresses. For that reason, it will normally not be useful to present a scenario in an easy configuration and then systematically repeat the presentation at increasing levels of difficulty, as might be done in many tracking tasks. Therefore, the difficulty ratings for different scenarios should be generally consistent.

Scenario Customization

When an easy and difficult prototype has been authored for a particular situation, automated processes can then generate a custom version, scaled to a desired intermediate level of difficulty for the next exercise. This can be done by a 'tweening' (Day, 1995) process, very similar to that done in image processing (more specifically image morphing) to produce a graphical image that represents some intermediate image *between* two key frames. Here, the two extreme instances created by the domain expert represent the key frames, and we wish to produce an intermediate composition that lies between them, the exact placement between the extremes being a function of the learner's proficiency relative to the difficulty levels of the two defined cases.

Figure 4 suggests the procedure to be used. It illustrates how a single attribute, the heading of one aircraft, would be adjusted while producing a scenario at difficulty level 7. In this example the easiest instance of the situation is classified at level 3 and the most difficult at level 9. Because the desired difficulty is 2/3 of the distance between the two extreme cases, we adjust the aircraft's heading by 2/3 of the difference between the two extreme cases, yielding a heading of 50 degrees. Appendix B presents the tweening formula to be used, and Appendix C presents a more complete example.

It should be noted that this tweening process correctly maintains relative values as well as absolute ones. Thus, if a scenario is made more difficult by the proximity of two attribute values, the proximity established in the adapted scenario will correctly correspond with

[4] We will say that there are two versions of a single scenario, thus the word 'scenario' refers to a situation, while the different versions present differing levels of difficulty for that situation.

the intended difficulty level. It should also be emphasized that the tweening process does not just deal with attributes that can be represented graphically. Attributes such as velocity, temperature, and wind speed will be adjusted as well as observable characteristics of the agents in a scenario. This process appears to offer considerable potential for generating scenarios of desired difficulty.

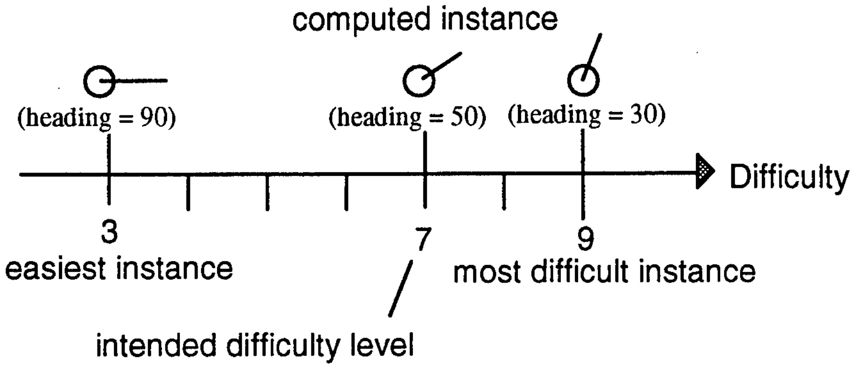


Figure 4. Tweening the Heading of a Track.

Computing Proficiency of Performance

The author will provide one or two complementary and domain-specific rule sets to express task proficiency implications of various conditions:

1) a scenario-specific rule set that specifies conditions that support various proficiency conclusions; and
2) a task-specific rule set that specifies conditions that support various proficiency conclusions.

The scenario-specific rules apply to a particular situation, such as *off-course airliner*; the task-specific rules apply to the general scenario domain, such as *air traffic control.* Note that the scenario-specific rule set applies to all instances of the scenario; there is not a need to produce rules that apply to individual instances.

These two rule sets may overlap, and either one but not both may be empty. For some tasks, it may be possible to completely specify proficiency conditions in the task rule set, thereby eliminating any need to produce scenario-specific rule sets. For some tasks, there may not be any useful rules that apply to all scenarios, therefore the task-specific rule set is empty. In general, there will be some rules that can be formulated that apply across all scenarios, yet each scenario will require some additional rules express the implications of particular conditions within the scenario. The proficiency rules are cast in terms of the authored *status measures*, as described above. Appendix D provides detailed specifications of the rules.

Acquiring Proficiency Scores From Experts

During tutor development, one or more experts perform the prototype scenarios, and their proficiency scores are computed just as they are for learners. These scores are then used as the reference against which the student proficiency scores are compared.

The Scenario Selection Process

We can now consider the entirety of the process by which a system could assess learner proficiency and set a desired difficulty level for the next exercise. Figure 5 illustrates the process.

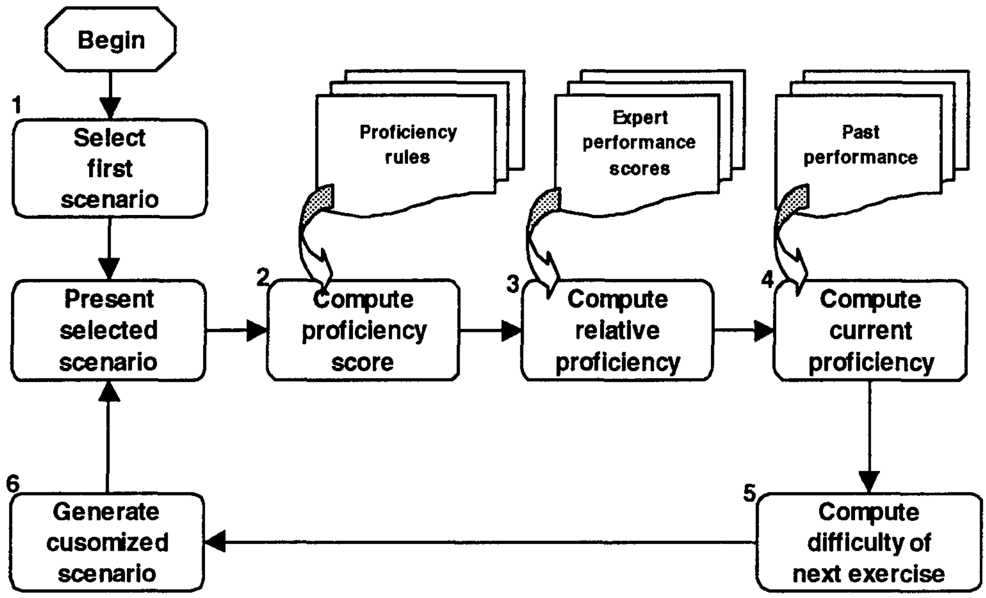


Figure 5. Proficiency Assessment and Exercise Generation Process.

The general process is as follows:

1. The initial scenario is selected, as specified by the instructor, and presented.
2. Based upon the authored proficiency rules, a proficiency on the *current* exercise is computed.
3. Based upon previously recorded performance of experts, a *relative proficiency* measure is computed.
4. Using a weighted moving average, a *current level of proficiency* is computed.
5. Based upon the current level of proficiency, the difficulty of the next exercise is established.
6. A scenario instance of the desired level of difficulty is generated and presented.

Computing an Exercise Proficiency Score

As the student or team works an exercise, the proficiency rules are evaluated, yielding a measure for the exercise when it is completed. This is identical to the process by which the expert scores are determined on the easy and difficult prototype scenarios.

Computing Relative Proficiency

Having a raw proficiency score for an individual or team on a just-completed exercise, a measure is computed relative to how well an expert *would have performed* that exercise. To determine this, the previously recorded scores of expert performance are interpolated to derive an expected score on the just-completed scenario. Then the learner's score is divided by the projected expert score to yield a relative measure.

Computing Current Proficiency

Using the relative proficiency measure for the just-completed exercise, and the individual's proficiency on previous exercises, a moving average would be computed to express the learner's current proficiency level. This function could be adjustable by the instructional organization, and might take the form

$$p' = Ap_1 + Bp_n$$

where

p' is the computed proficiency level for the individual, termed the *current* proficiency level

A and B are weights that total 1

p_1 is the proficiency of the individual on the previous exercise

p_n is the average proficiency over the prior n exercises

If the number of exercises completed is less than n, then p_n is the average proficiency over the completed exercises (and if the number of exercises completed is 0 then A = 1 and B = 0).

Establishing the Difficulty of the Next Exercise

Following the first exercise, and all others, the instructional system will determine the appropriate difficulty level for the next exercise, based upon the individual's current proficiency level and current difficulty level. The exact parameters to employ for this are a matter of some experimentation. The following scheme appears to be reasonable.

Learner's current Proficiency Level	Change in Difficulty of next exercise
<= .4	-2
>.4 - .6	-1
>.6 -.8	0
.> .8	+1

Thus a learner given a first exercise at difficulty level 5 and completing it at 80% of expert performance would receive another exercise at difficulty level 5. If performance on that subsequent problem brought the learner's running proficiency level down to .58, then the difficulty level would be dropped to 4, whereas a proficiency increase to .85 would call for the next exercise at difficulty level 6.

The particular scheme for setting difficulty level will be one of the instructional preferences editable at the instructional site, for this may also be heavily influenced by the policies and requirements of the organization. Qualification standards, for example, may require that learners experience certain difficult scenarios, regardless of their proficiency. Or, the training organization may find that motivation and learning is increased when the difficulty level is increased or decreased from the particular scheme outlined above.

Expert Tutoring

In addition to overseeing the generation of scenarios, the intelligent tutor will be concerned with providing within-exercise support and after-exercise debriefs. This will be done using Performance rules authored for the domain. As with the Proficiency rules, it may be possible to specify expert performance at the domain level, but in general the developer will also provide scenario-specific rules to express expert performance.

Control of instruction will be shared by the learner and the system. There will be four modes of instructional interaction provided during exercises. In order of increasing interaction with the learner, they are:

1. a proficiency evaluation mode, in which the training system silently evaluates individual performance;
2. an on-demand mode, in which instructional content is provided only upon request;
3. a critical-condition mode, in which the instructional system detects and remediates critical task conditions; and
4. a guided performance mode, in which the training system interacts with the learner at each critical decision point.

In all modes the instructional system will continually evaluate the proficiency rules to maintain a measure of individual ability. In general, the learner may call upon the system for assistance at any time except during performance evaluation exercises. There will also be some simple means by which the instructional organization can constrain use of the instructional functions, thereby limiting on-demand assistance in various ways.

In the on-demand mode, the system will evaluate the task condition existing at the time of the request for consultation, based upon the Proficiency rules, i.e., it will describe and attempt to remediate any critical conditions which the learner's past decisions have created. In the critical-condition mode of instruction, the system will intercede whenever a critical condition is sensed, while in the guided mode the system interacts each time a Performance rule indicates that a decision is required.

When instruction is provided during an exercise the system will -

- freeze the simulation;
- provide a situation assessment based upon the rules of expertise in relation to the current situation, and
- advise the learner what to do and why to do it. After considering the instructional advice, the learner resumes the exercise by clicking the mouse.

<u>After Exercise Instruction</u>. Following completion of an exercise, the learner may request either or both of the following:

- Debrief (replay). The system will play back the exercise just completed by the learner, providing expert analysis of the learner's performance based upon the proficiency rules. The learner may run this demonstration in either of two modes: 1) an 'automatic-stop' mode in which the system automatically freezes with the elicitation of each new performance rule, and continues when the learner clicks the mouse, or 2) a 'continuous run' mode in which the demonstration runs in real time, with each activated performance rule displaying until replaced by another. It may be that screen space will permit performance rules to be added to a scrolling list.

- Expert Performance. The system will demonstrate expert performance on the just completed scenario, using that instance that most closely matches the difficulty level at which the learner experienced the situation. During this demonstration, the system will continuously assess the performance rules, displaying those that pertain at each instant. This instructional function will also be available in either the automatic-stop or continuous-run modes.

VII. Conclusions

At the highest level there is considerable congruence of goals and functions among different scenario tasks. This bodes extremely well for the development of a generalized authoring and instructional system, for it allows one to anticipate the kinds of content and processes that will be required in both the authoring environment and the instructional delivery environment.

Just as clearly, however, different scenario tasks bring in completely different domain-specific content, including the nature of the agents, the appearance of the world presented to the learner, the subset of the physical world in which the learner must function, and the repertoire of actions available. Consequently, it is the task of the author to 1) construct a simulation of that domain to whatever level of realism is desired, 2) produce expressions that reflect the status of the simulation as the learner works, 3) express conditions that *reflect* proficiency of performance, and 4) formulate rules that collectively *prescribe* proficient performance.

Past experience suggests that computer-literate individuals can do these tasks although representing expertise in a rule base can be exceedingly difficult in some cases, particularly if the experts have difficulty telling us explicitly what they do. In these cases it is likely that one still could assemble a reasonable set of proficiency indicators, for these deal with specific cases rather than general prescriptions. Thus, the worst case appears to be one in which the tutoring system would not have the capacity to demonstrate and debrief, but would instead generate and present scenarios for practice, with the possibility of conversing about proficiency issues detected during practice.

With this worst case in mind, we conclude the report by revisiting the training systems and training research surveyed previously. For each such project, we offer a brief synopsis of the extent to which the provisional authoring system might meet have met its needs, had it been used as the development environment.

GT-AAWC. Past experience proves the feasibility of simulating the CIC environment with the authoring tools already in place. The rules of engagement used in GT-AAWC, along with the windows of opportunity, could easily be cast in the proficiency and performance rule form outlined here, and the prioritized feedback scheme could easily be implemented as well.

DDT/LC2. The domain appears to be entirely amenable to implementation in a generalized authoring system. The orientation and operations phases used in DDT/LC2 would compress into the practice phase in the generalized instructional routine, and the debriefing phase would remain distinct. The scenario generation scheme of the DDT/LC2 system appears to be reproducible, although some change might be required to force the scenario generator to stay with the same scenario until expert performance is attained.

DEFTT and TADMUS DSS. Considering the extensive instrumentation and networking of the DEFTT facility, the use of a generalized authoring environment may be inappropriate. As mentioned above, however, the key decision making task of the CO and TAO is clearly one that could be addressed in a generalized environment. Augmenting the simulation of the real PPI with a DSS introduces no particular problems in this case, since the graphics of the DSS are not exotic.

PC-IMAT. Integration of the extensive data bases, oceanographic processes, and graphics display generation used in PC-IMAT do not present a theoretical problem in a generalized authoring environment, but they probably present a practical obstacle related to compatibility of operating system requirements, and so on. The experimentation with complete user control versus a student model could be fully implemented in a generalized setting.

TRIDENT, 21A43, and TRIDENT MK trainers. The extensive use of real equipment in these trainers essentially places them outside the realm that could be implemented in a generalized environment.

Run (Fire Commander and Zoo Keeper). There appear to be no requirements of this learning environment that could not be implemented.

Research in teams working under stress (Hall, Dwyer, Cannon-Bowers, Salas, and Volpe). The extensive analysis of scenario factors performed by these researchers would lead directly to their specification and construction of low stress (easy) and high stress (difficult) scenarios. The tweening process would generate scenarios at appropriate levels in between. The occurrence or non-occurrence of events leading to ambiguity might pose a problem for the tweening process, since these events appear to be all or nothing. Perhaps the time at which the event, such as loss of electronic emissions, could be used to control the stress level. If not, separate scenarios would be required to involve this new factor.

Damage Control. The domain of damage control appears, superficially, to be amenable to implementation via proficiency and performance rules. If task-level rules cannot effectively be constructed, it would seem that scenario-level rules could be.

Shipboard Instructor Training and Support (SITS). Many of the goals of this work could be met in a generalized development environment. The design described in this report did not explicitly address a methodology for formally linking training objectives to scenario events, however it appears that the explicit and formal process would be used when constructing the prototype scenarios. Thus, the approach described here would not automatically involve training objectives, as appears to be the objective of SITS, but would permit such involvement as an external operation.

References

Brecke, F. H. and Garcia, S. K. (1995). *Training methodology for logistic decision making* (AF/HR-TR-1995-0098). Brooks AFB, TX: Armstrong Laboratory.

Day, A. W. (1995). *Making and using animations for science education.* Paper presented at the Computers in University Biological Education virtual conference, Liverpool.

Dwyer, D.J., Hall, J.K., Volpe, C., Cannon-Bowers, J.A., and Salas, E. (1992). *A Performance Assessment Task for Examining Tactical Decision Making Under Stress*, (Report 92-002), Orlando: Naval Training Systems Center.

Hall, J. K., Dwyer, D. J., Cannon-Bowers, J. A., Salas, E., & Volpe, C. E. (1993). Toward assessing team tactical decision making under stress: The development of a methodology for structuring team training scenarios. *Proceedings of the 15th Annual Interservice/Industry Training Systems Conference* (pp. 87-98), Washington, DC: National Security Industrial Association.

Hunt, E., Joslyn, S, and Sandquist, T. (1996). *Evaluating Individual Differences in Response to Emergency Situations.* (ONR Final Report) Seattle: University of Washington.

Inzana, C.M. (1993). *The Effects of Preparatory Information on Enhancing Performance Under Stress*, Orlando: University of Central Florida, Thesis.

Kaempf, G. L., Wolf, S., Thordsen, M. L., & Klein, G. (1992). *Decisionmaking in the AEGIS combat information center*, (NOCCSC Technical Report), Fairborn, OH: Klein Associates Inc.

Kirlik, A., Fisk, A. D., and Walker, N. Feedback Augmentation and Part-Task Practice in Training Dynamic Decision Making Skills, in E. Salas and J. Cannon-Bowers (Eds.), *Decision Making in Complex* Environments*: Implications for Training and Simulation.* Washington D.C.: American Psychological Association.

Klein, G. (1992). *Decisionmaking in complex military environments*, (NCCOSC Final Contract Summary Report), Fairborn, OH: Klein Associates Inc.

McCarthy, J. E., Pacheco, S., Banta, G. H., Wayne, J. L., and Coleman, D. S. (1994). *The radar system controller intelligent training aid.* Proceedings of the 16th Annual Interservice/Industry Training Systems Conference. Orlando, FL: National Security Industrial Association.

Miller, T. E., Wolf, S. P., Thordsen, M. L., and Klein, G. (1992). *A decision-centered approach to storyboarding Anti-Air Warfare Interfaces*, (NCCOSC Technical Report, Task 3), Fairborn, OH: Klein Associates Inc.

Munro, A. (1994). Authoring Interactive graphical models. In T. de Jong, D. M. Towne, and H. Spada (Eds.), *The Use of Computer Models for Explication, Analysis, and Experiential Learning*. Springer Verlag.

Rothrock, L.R. (1995). *Performance measures and outcome analyses of dynamic decision making in real-time supervisory control.* Unpublished doctoral dissertation. School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Stretton, M. L. and Lackie, J. H. (1996). Shipboard Instructor Training and Support: Scenario Development functional architecture overview. (SITS program technical review paper).

Tannenbaum, S. I., Beard, R. L., and Salas, E. Team building and its influence on team effectiveness: An examination of conceptual and empirical developments. In K. Kelly (Ed.), *Issue, theory, and research in industrial/organizational psychology*. Amsterdam: Elsevier.

Towne, D. M. (1994). Model based simulations for instruction and learning, in *Proceedings, Delta '94 Conference: Telematics for Education and Training*. Dusseldorf.

Towne, D. M. (1995). *A Configurable Task Environment for Learning Research*, (ONR Final Report), Los Angeles: Behavioral Technology Laboratories, University of Southern California.

Towne, D. M. (1997). Intelligent instruction of discrete dynamic decision making tasks, (ONR Proposal), Los Angeles: Behavioral Technology Laboratories, University of Southern California.

Travillian, K.K. (1993). *Cross-Training: The Effects of Positional Rotation and Workload on Affective Reactions, Team Processes and Team Performance*, Orlando: University of Central Florida, Thesis.

Travillian, K.K., Volpe, C.E., Cannon-Bowers, J.A., and Salas, E. (1993). Cross-training highly interdependent teams: Effects on team process and team performance, in *Proceedings of the 37th Annual Human Factors and Ergonomics Society Conference.*

Weaver, J. L., and Morgan, B. B. Jr. (1993). Team Decision Making in the Command Information Center: Development of a Low-Fidelity Team Decision Making Task for Assessing the Effects of Teamwork Stressors. Orlando: Naval Training Systems Center.

Zachary, W. W., Zaklad, A. L., Hicinbothom, J. H., Ryder, J. M., Purcell, J. A., & Wherry, R. J. (1991). *COGNET representation of tactical decision-making in ship-based anti-air warfare* (NCCOSC technical report). San Diego, CA: Naval Command, Control and Ocean Surveillance Center.

Appendix A

Systems Reviewed by Stretton and Lackie (Stretton and Lackie, 1996, ibid.)

20BS FFG-7 Trainer;

AN/SQQ-89(V) On-Board Trainer (OBT);

AN/SQ5-53A EC-16 Interim On-Board Trainer (IOBT);

ACTS, Baseline 4, Phase 2;

ACTS, Baseline 5, Phase 3;

Aimpoint;

Army Synthetic Forces (ARMYSF)/Modular Semi-Automated Forces (MODSAF);

ASW Evaluator Maneuvering Training Aid (ASWETA II);

Automated Construction Exercise Database (ACED);

Batman and Robin;

Battle Force Team Training (BFTT) System;

Interactive Tactical Environment Management System (ITEMS);

Navy Air Synthetic Forces (NAVAIRSF);

OBT Trainer Control Device (TCD);

Radar System Controller Intelligent Training Aid (RSC ITA);

Scenario Generator;

Tactical Advanced Combat Direction and Electronic Warfare (TACDEW);

Environmental Generation and Control System (EGCS);

Tactical Aircraft Mission Planning System, 5. series;

Tactical Aircraft Mission Planning System, 6. series;

Tactical Advanced Simulated Warfare Integrated Trainer (TASWIT);

Tactical Exercise Force Laydown (TEFL);

Training Event Design System (TREDS).

Appendix B

Adapting A Scenario Attribute to Specified Difficulty Level

The formula is for adjusting an attribute, a, in terms of

1) its values in the easier and more difficult scenario instances,
2) the difficulty levels of the easier and more difficult scenario instances, and
3) the desired difficulty of the adapted scenario.

The formula is

$$a' = a_e + (a_d - a_e)(D - i_e) / (i_d - i_e)$$

where

a' is the adjusted value of the attribute

a_e is the value of the attribute in the easier instance

a_d is the value of the attribute in the more difficult instance

D is the difficulty level of the scenario being produced

i_d is the difficulty level of the more difficult scenario instance

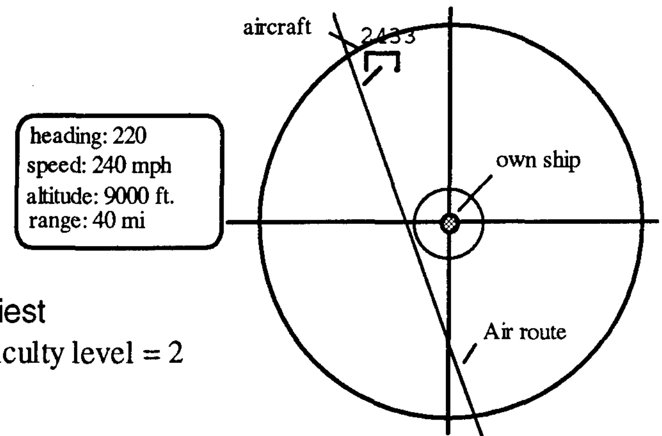
i_e is the difficulty level of the easier scenario instance

Suppose we wish to produce a scenario at difficulty level 7, working from a situation whose easier instance is rated at level 3 and more difficult instance is rated at 9. For an attribute whose value is 50 in the easier instance and 20 in the more difficult instance, its value in the adapted instance, a', would be computed as follows:

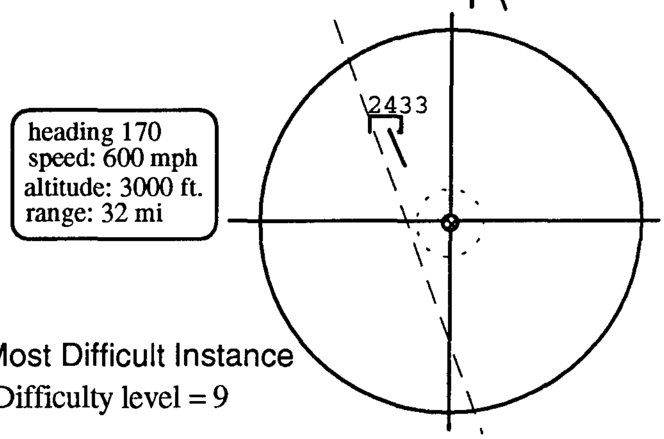
$$a' = 50 + (20 - 50)(7 - 3) / (9 - 3) = 50 + (-30)(4)/6 = 50 - 20 = 30$$

Appendix C
Scenario Adaptation Example

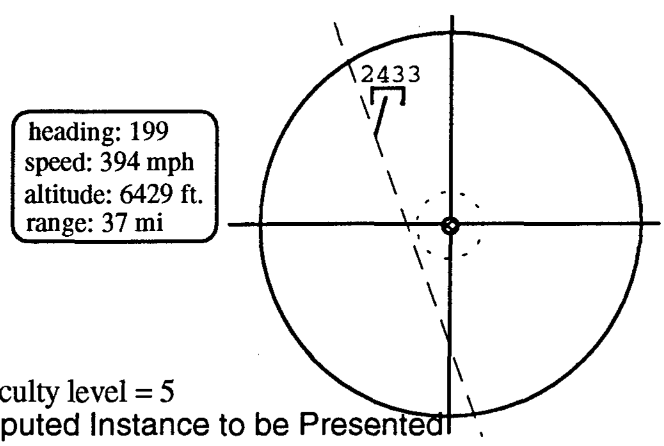
The figures illustrate a simple CIC situation in which a commercial airliner has veered off course, a course that passes nearly over own ship. In the easy instance the airliner is far off course, traveling slowly and at high altitude. This has been assigned a difficulty level of 2. In the most difficult instance, at difficulty level 9, the airliner is traveling faster, lower, and is heading more toward the ship, as well as being closer to the decision maker at the start of the exercise. The third figure is that which would be produced by the scenario customization module to correspond with a difficulty level 5.



Easiest
Difficulty level = 2



Most Difficult Instance
Difficulty level = 9



Difficulty level = 5
Computed Instance to be Presented

Appendix D

Syntax of Performance and Proficiency Rules

The entirety of the intelligent instructional system, including the selection and adaptation of scenarios, the instructional interactions, and the evaluation of learner proficiency, depend upon automated analysis of the individual's performance and of the situational conditions that result therefrom. All of this will be supported by capturing and interpreting two kinds of rules:

1. *Performance* rules, expressing how the decision making task should be performed; and
2. *Proficiency* rules, expressing the proficiency implications of various critical situations.

Both types of rules will be supplied by the domain experts, in the form of conditional (if-then) statements. The rules will be entered to a special authoring system that will accept, check and file the rules. The rules will be in a conventional conditional syntax, e.g.,
If <some condition> then <some outcome>.

Condition Part. The *condition* part of the rules will express any particular combination of task conditions that pertain during an exercise. Example condition parts of rules (stated in English) are:

if range_of_nearst_hostile < 3
or
if time_since_level3_warning > 5

To implement such rules, the author would provide expressions or procedures for maintaining the values of the variables range_of_nearest_hostile and time_since_level3_warning.

Outcome Part. For Performance rules, the outcome part expresses the decision that should be made in that condition. This will provide the recommended decision in that condition. For example

if <some condition> then *illuminate.*
or
if <some condition> then *towerCheck*

Also, attached to each Performance rule will be a *rationale* for the recommended action. This will be communicated to the learner whenever the instructional system provides a recommendation.

For Proficiency rules, the outcome part expresses the proficiency implications of the condition. For example, if some seriously threatening condition exists in a particular scenario, then that constitutes evidence of a proficiency deficit. The outcome part of these rules is simply a positive or negative number that indicates the seriousness of the condition, and whether it is a desirable or undesirable condition. This is used by the instructional system to maintain a measure of the individual's proficiency.

Some hypothetical performance rules are:

```
if NearestHostile_range < 15 and timeSinceWarning > 3 then illuminate
if nearestHostile_speed > 600 and nearestHostile_altitude < 2000 then warn2
if nearestHostile_range < 10 and nearestHostile_bearing < 30 then fire
```

Some hypothetical Proficiency rules are:

```
if timeSinceWarning >3 and nearestUnknown_range < 2 then -3
if nearestHostile_range < 5 and nearestHostile_angleOff < 20 then -2
```

While the syntax of these rules is generally consistent with that used in many computer programming languages, we would expect that non programmers could learn to formulate and enter them relatively easily.